**Install Jupyter Notebook**

**Create a project directory and install Jupyter Notebook**

1. mkdir ~/myprojectdir
2. cd ~/myprojectdir
3. pip install jupyter
4. cd ..
5. vi .bashrc

In the vi editor, you can add the following line at the end

export PATH=”/home/<userid>/.local/bin/jupyter:$PATH”

You now have everything you need to run Jupyter Notebook! To run it, execute the following command:

1. jupyter notebook

A log of the activities of the Jupyter Notebook will be printed to the terminal. When you run Jupyter Notebook, it runs on a specific port number. The first notebook you run will usually use port 8888. To check the specific port number Jupyter Notebook is running on, refer to the output of the command used to start it:

Output

[I 21:23:21.198 NotebookApp] Writing notebook server cookie secret to /run/user/1001/jupyter/notebook\_cookie\_secret

[I 21:23:21.361 NotebookApp] Serving notebooks from local directory: /home/sammy/myprojectdir

[I 21:23:21.361 NotebookApp] The Jupyter Notebook is running at:

[I 21:23:21.361 NotebookApp] http://localhost:8888/?token=1fefa6ab49a498a3f37c959404f7baf16b9a2eda3eaa6d72

[I 21:23:21.361 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).

[W 21:23:21.361 NotebookApp] No web browser found: could not locate runnable browser.

[C 21:23:21.361 NotebookApp]

Copy/paste this URL into your browser when you connect for the first time,

to login with a token:

http://localhost:8888/?token=1fefa6ab49a498a3f37c959404f7baf16b9a2eda3eaa6d72

At this point, you can keep the SSH connection open and keep Jupyter Notebook running or can exit the app and re-run it once you set up SSH tunneling. Let’s keep it simple and stop the Jupyter Notebook process. We will run it again once we have SSH tunneling working. To stop the Jupyter Notebook process, press CTRL+C, type Y, and hit ENTER to confirm. The following will be displayed:

Output

[C 21:28:28.512 NotebookApp] Shutdown confirmed

[I 21:28:28.512 NotebookApp] Shutting down 0 kernels

We’ll now set up an SSH tunnel so that we can access the notebook.

**SSH Tunneling**

In this section we will learn how to connect to the Jupyter Notebook web interface using SSH tunneling. Since Jupyter Notebook will run on a specific port on the server (such as :8888, :8889 etc.), SSH tunneling enables you to connect to the server’s port securely.

The next two subsections describe how to create an SSH tunnel from 1) a Mac or Linux and 2) Windows. Please refer to the subsection for your local computer.

[**SSH Tunneling with a Mac or Linux**](https://www.digitalocean.com/community/tutorials/how-to-set-up-jupyter-notebook-with-python-3-on-debian-9#ssh-tunneling-with-a-mac-or-linux)

If you are using a Mac or Linux, the steps for creating an SSH tunnel are similar to using SSH to log in to your remote server, except that there are additional parameters in the ssh command. This subsection will outline the additional parameters needed in the ssh command to tunnel successfully.

SSH tunneling can be done by running the following SSH command in a new local terminal window:

1. ssh -L 8888:localhost:8888 your\_server\_username@your\_server\_ip

Copy

The ssh command opens an SSH connection, but -L specifies that the given port on the local (client) host is to be forwarded to the given host and port on the remote side (server). This means that whatever is running on the second port number (e.g. 8888) on the server will appear on the first port number (e.g. 8888) on your local computer.

Optionally change port 8888 to one of your choosing to avoid using a port already in use by another process.

server\_username is your username (e.g. sammy) on the server which you created and your\_server\_ip is the IP address of your server.

For example, for the username sammy and the server address 203.0.113.0, the command would be:

1. ssh -L 8888:localhost:8888 sammy@203.0.113.0

Copy

If no error shows up after running the ssh -L command, you can move into your programming environment and run Jupyter Notebook:

1. jupyter notebook

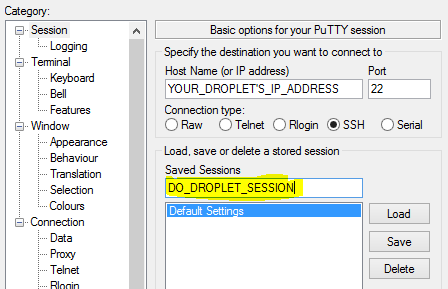
Copy

You’ll receive output with a URL. From a web browser on your local machine, open the Jupyter Notebook web interface with the URL that starts with http://localhost:8888. Ensure that the token number is included, or enter the token number string when prompted at http://localhost:8888.

[**SSH Tunneling with Windows and Putty**](https://www.digitalocean.com/community/tutorials/how-to-set-up-jupyter-notebook-with-python-3-on-debian-9#ssh-tunneling-with-windows-and-putty)

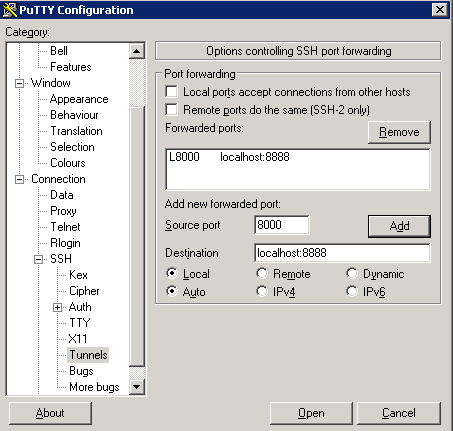
If you are using Windows, you can create an SSH tunnel using [Putty](https://www.putty.org/).

First, enter the server URL or IP address as the hostname as shown:



Next, click **SSH** on the bottom of the left pane to expand the menu, and then click **Tunnels**. Enter the local port number to use to access Jupyter on your local machine. Choose 8000 or greater to avoid ports used by other services, and set the destination as localhost:8888 where :8888 is the number of the port that Jupyter Notebook is running on.

Now click the **Add** button, and the ports should appear in the **Forwarded ports** list:



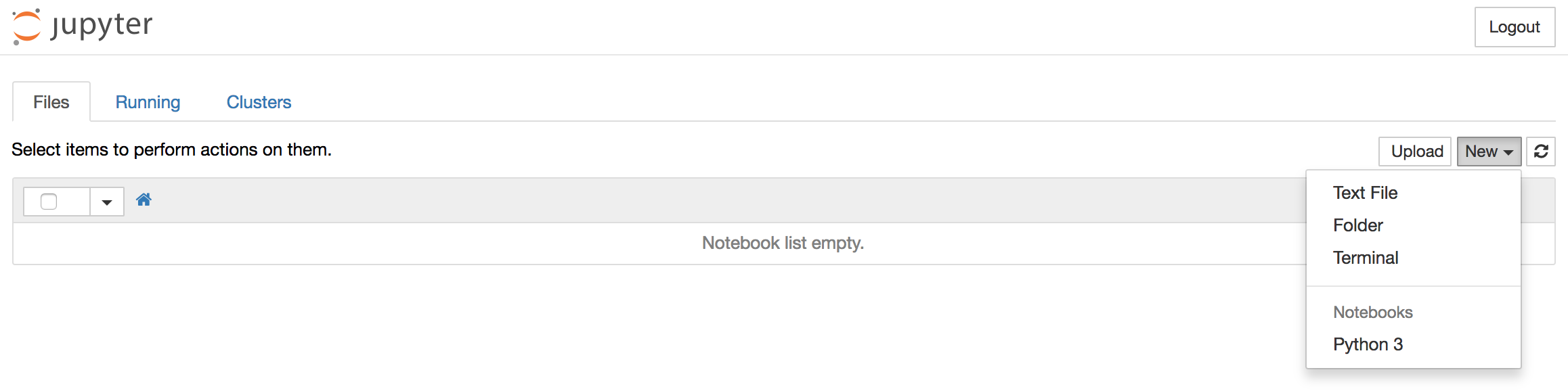
Finally, click the **Open** button to connect to the server via SSH and tunnel the desired ports. Navigate to http://localhost:8000 (or whatever port you chose) in a web browser to connect to Jupyter Notebook running on the server. Ensure that the token number is included, or enter the token number string when prompted at http://localhost:8000.

[**Step 6 — Using Jupyter Notebook**](https://www.digitalocean.com/community/tutorials/how-to-set-up-jupyter-notebook-with-python-3-on-debian-9#step-6-using-jupyter-notebook)

This section goes over the basics of using Jupyter Notebook. If you don’t currently have Jupyter Notebook running, start it with the jupyter notebook command.

You should now be connected to it using a web browser. Jupyter Notebook is very powerful and has many features. This section will outline a few of the basic features to get you started using the notebook. Jupyter Notebook will show all of the files and folders in the directory it is run from, so when you’re working on a project make sure to start it from the project directory.

To create a new notebook file, select **New** > **Python 3** from the top right pull-down menu:



This will open a notebook. We can now run Python code in the cell or change the cell to markdown. For example, change the first cell to accept Markdown by clicking **Cell** > **Cell Type** > **Markdown** from the top navigation bar. We can now write notes using Markdown and even include equations written in [LaTeX](https://www.latex-project.org/) by putting them between the $$ symbols. For example, type the following into the cell after changing it to markdown:

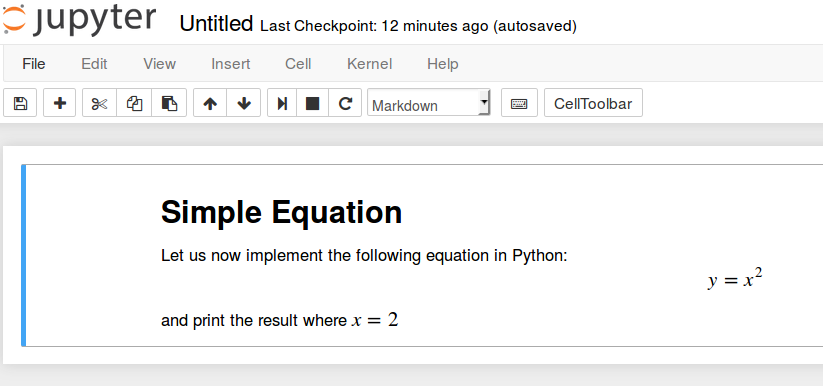
# Simple Equation

Let us now implement the following equation:

$$ y = x^2$$

where $x = 2$

To turn the markdown into rich text, press CTRL+ENTER, and the following should be the results:



You can use the markdown cells to make notes and document your code. Let’s implement that simple equation and print the result. Click on the top cell, then press ALT+ENTER to add a cell below it. Enter the following code in the new cell.

x = 2

y = x\*\*2

print(y)

Copy

To run the code, press CTRL+ENTER. You’ll receive the following results:

